

基于 FPGA 的 JPEG 压缩编码设计与实现*

余磊, 李磊, 崔建明, 陈新华

(山东科技大学 信息科学与工程学院, 山东 青岛 266590)

摘要: 利用 FPGA 可以并行处理数据的优点, 设计出了一种 JPEG 压缩编码电路。并在尽量保证图像质量的前提下, 对 JPEG 的传统编码过程进行优化与调整, 简化了编码电路, 提高了编码效率, 并生成独立 IP 核, 方便调用。实验结果表明, 设计的编码电路完全达到了预期目的。

关键词: JPEG; FPGA; 压缩编码; IP

中图分类号: TP919.81

文献标识码: A

文章编号: 1674-7720(2012)21-0023-03

Design and implementation of JPEG compression based on FPGA

Yu Lei, Li Lei, Cui Jianming, Chen Xinhua

(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China)

Abstract: Based on the advantage that FPGA can parallel processing data, the paper designs a JPEG compression coding circuit. In addition, the paper optimizes and adjusts the JPEG encoding process on the condition that it can ensure the image quality, making the coding circuit simple and efficient, and generate an independent IP core, which is convenient to call. The experimental results show that the encoding circuit design reaches the intended purpose.

Key words: JPEG; FPGA; compression; IP

JPEG 编码是一种被广泛使用的图形压缩标准, 由于其压缩率高、图像质量高且无版权专利费而深受大家喜爱, 因此对于 JPEG 压缩编码的研究就从来没有停止过。虽然之前已经有了一些实现方式, 如纯软件、DSP 与专用集成电路方式, 但都存在一些缺点, 例如软件方式压缩速度较慢、DSP 成本较高、专用芯片灵活性不够以及不易对压缩过程进行调整等。本文采用的是一种基于 FPGA 的压缩方法, 它充分利用了 FPGA 灵活、高效的特点, 再配合 Nios II 实现对过程的控制, 并对压缩算法进行了一些改进与优化。

设计使用 DE2-70 集成开发板作为设计平台。FPGA 芯片为 ALTERA 公司的 EP2C70F896C6N 芯片, 它具有 32 MB×2 SDRAM、2 MB SSRAM、68 416 个 LE 单元以及丰富的外设资源。在设计中, 数据的流动由 Nios II 进行控制, 同时它也负责图片的读入与输出, 而压缩过程中每一步都是由相对独立的处理模块完成。

1 编码过程

根据国际标准化组织给出的 JPEG 压缩标准流程

为: 对原始位图数据首先进行 RGB 到 YUV 的转换 (即得到亮度与色度信息); 并对每个 8×8 像素块做离散余弦变换 (DCT); 然后对所得结果做量化; 接着进行 Z 型数据重排; 最后是熵编码。编码过程如图 1 所示。

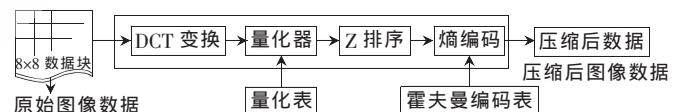


图 1 编码过程框图

1.1 亮度空间转化

压缩编码的对象是由 YUV 表示的亮度、色度数据, 需对原始 RGB 数据做变换。由于本设计中只要求得到亮度图像, 因此转换出了亮度信息 (即 Y 分量)。计算 Y 分量的表达式为:

$$Y = (0.275 \times R) + (0.504 \times G) + (0.098 \times B) + 16$$

由于上式中存在小数部分, 在硬件计算中比较麻烦, 所以改为整数定点计算。计算式如下:

$$(Y \ll 10) = (12'h107 \times R) + (12'h204 \times G) + (12'h64 \times B) + 20'h4000$$

即先将系数同时向右移动 10 bit 变为整数, 计算完

* 基金项目: 青岛科技局科技计划项 (07-2-3-1jc)

成后再将结果左移 10 bit。这样可以避免小数运算。在计算完成后,对结果截取了高 8 bit,判定是否溢出,如果小于 0,则将结果赋予 0,大于 255 时,赋予最大值 255。

1.2 DCT 变换

根据离散余弦变换定义,对一个 8×8 单元正向 DCT 变换的数学表达式为:

$$F(u, v) = \frac{1}{4} C(u) C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cdot \cos \frac{(2x+1)u\pi}{16} \cdot \cos \frac{(2y+1)v\pi}{16} \right]$$

其中, $x, y, u, v = 0, 1, 2, \dots, 7, C(u), C(v) = \begin{cases} 1/\sqrt{2}, u, v = 0 \\ 1, \text{others} \end{cases}$,

这是标准的 2 维 DCT 变换实现方法。如果直接按照上式进行计算,那么对于一个 8×8 像素块将会需要大概 4 096 次乘法运算和 4 096 次加法运算,它将拖慢整个系统的处理时间。因此在设计中使用如下一维方式计算 DCT 变换:

$$Y(u) = \frac{c(u)}{2} \sum_{x=0}^7 X(x) \cdot \cos \frac{(2x+1)u\pi}{16}$$

$$Y(u, v) = \frac{c(v)}{2} \sum_{y=0}^7 Y(u) \cdot \cos \frac{(2y+1)v\pi}{16}$$

将原本的二维 DCT 变换改为 2 次一维 DCT 运算。在计算中可以先对 8×8 块中的 8 行分别进行变化并将结果暂存,然后再对暂存的数据中的 8 列分别再次进行 DCT 运算,然后输出结果。并且在单次 DCT 运算中还存在以下规律:

$$\begin{bmatrix} y_0 \\ y_2 \\ y_4 \\ y_6 \end{bmatrix} = \begin{bmatrix} c_0 & c_0 & c_0 & c_0 \\ c_2 & -c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} \begin{bmatrix} x_0+x_7 \\ x_1+x_6 \\ x_2+x_5 \\ x_3+x_4 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_3 \\ y_5 \\ y_7 \end{bmatrix} = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ -c_3 & c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & -c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \begin{bmatrix} x_0-x_7 \\ x_1-x_6 \\ x_2-x_5 \\ x_3-x_4 \end{bmatrix}$$

其中:

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} \cos(4\pi/16) \\ \cos(\pi/16) \\ \cos(2\pi/16) \\ \cos(3\pi/16) \\ \cos(4\pi/16) \\ \cos(5\pi/16) \\ \cos(6\pi/16) \\ \cos(7\pi/16) \end{bmatrix}$$

由于所有 c_n 的余弦值都可以预先计算好,并为了可以使用定点乘法器,因此将它们值扩大 2^{16} 倍后代入上式。这样,在计算完成后进行一次 16 bit 的左移操作,就可以得到最终结果。

1.3 量化

在压缩过程中量化是十分重要的一步,通过变步长量化可以减小高频分量的大小,从而减少需编码的数据量。在此过程中,对各个分量除以对应的量化步长,对于一个 8×8 数据块来说,一共需要进行 64 次除法运算。

1.4 Z 型扫描

在 Z 型扫描过程中,会将原本 8×8 的数据块按照 ZigZag 形扫描重新排列,排列后会使得数据序列中可能为 0 的数据连续在一起,这有助于后面的熵编码。

1.5 熵编码

在熵编码中,采用霍夫曼编码。对于直流 DC 系数采用差分编码方式。具体方法为:使用本数据块中的 DC 系数减去上一块中的 DC 系数,从而得到需编码的差值,即:

$$DC(n) = DC(n)_{per} - DC(n-1)$$

对于交流 AC 系数则采用行程编码,方法为:先对除 DC 系数之外的 63 个 AC 系数从前开始依次向后查询,如果为 0 则继续下一个,直到非 0 系数出现,将此时前面已出现的 0 系数个数连同这个非 0 系数一同进行编码。如果有连续 16 个 0 系数出现,则将它编码为 (15, 0); 如果从某一个 0 系数之后的其他系数全为 0,则输出一个结束符 EOF 即可。

霍夫曼编码是一种基于概率统计的编码方式,将码流中出现频率较高的码字给予较短的编码,这样可以有效减小码流。在实际操作中,对于直流 DC 与交流 AC 系数的编码使用不同的霍夫曼编码表。

2 JPEG 编码 FPGA 实现

整个系统的组成使用了 IP 核+Nios II 的实现方法。

其中,IP 核以可以实现特定计算功能的模块所组成,而嵌入 Nios II 是为了能够方便的对 IP 核进行控制,以便完成原始图像输入、编码后数据输出。其整体原理框图如图 2 所示。

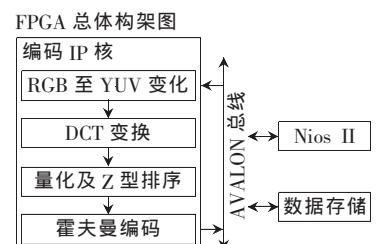


图 2 FPGA 整体组成框图

在整个过程中,位图图片首先被保存在片外存储器中,在 Nios II 的控制下依次取出 8×8 的数据块,之后,交给第一个处理模块中,图像被编码完成后,同样在 Nios II 控制下被送至指定存储区域。

2.1 RGB 至 YUV 转换模块

在 RGB 至 YUV 转换模块中,将完成图像的 RGB 至 YUV 转变。由于设计得到的图像只是亮度图像,因此设计中转化出了亮度信息(即 Y 分量)。模块的输入为像素点的 RGB 3 个分量值(每分量 8 bit),输出为此像素点的 Y(亮度)信息(8 bit)。其过程如图 3 所示。

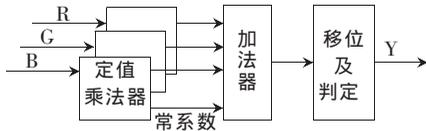


图3 RGB至Y的转换过程

设计使用到的3个定值乘法器使用Quartus中提供的3输入乘加器。

2.2 DCT变换模块

在整个压缩编码的过程中DCT变换尤为重要,它是整个压缩编码的基础,同时它的计算量也较大,因此正确、高效的DCT模块显得尤为重要。在DCT变换模块的设计上遵照1.2节的处理流程,其过程如图4所示。

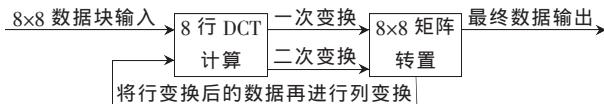


图4 DCT变化过程

按照1.2节介绍,可以把DCT变换分为两次独立的一维DCT变换。在开始时刻先是对 8×8 数据块中的8行分别做DCT计算。在Memory控制器下对数据行、列转置存储。然后再将转置后的 8×8 数据送入DCT计算块做运算。最后再将结果进行行、列转置存储。此时得到的数据即为最终的DCT变换结果。

2.3 量化及Z型编码

在量化及Z型编码过程中, 8×8 块的64个量化系数表与Z排序系数表已经提前存入片上ROM。处理时,先从ROM中取出量化系数并与之对应的数据做完量化运算(取模)后,再按照Z排序系数表中对应的存储地址,直接放入RAM中对应位置。过程如图5所示。



图5 量化及Z排列过程

2.4 熵编码

由于霍夫曼编码中对DC系数与AC系数使用的编码方式与使用和霍夫曼编码表不同,因此在设计中使用了两个相对独立的模块分别对DC与AC系数进行编码。系统流程如图6所示。

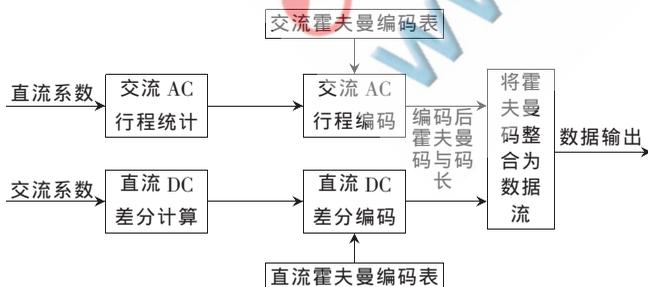


图6 霍夫曼编码过程

为提高速度,在直流与交流的编码过程中,数据只是做了必要的寄存,没有再将整个 8×8 大小的数据进行存储、读出。在AC的行程编码时,只要有非0信号出现(可以对此段进行编码时)就将数据给予后边的霍夫曼

编码部分作处理。在数据整合输出部分,只要集合够32 bit的有效数据就立即将之写入到存储器。不会让后一数据处理模块等待前一模块完全处理完 8×8 的数据而产生的等待延时,提高数据处理效率。

3 测试结果

下例是单独对 8×8 像素块的压缩对比,图7(a)是BMP格式的原始图像,大小为246 B,而图7(b)是压缩后的亮度JPG图像,其数据量只有18 B。



图7 BMP原始像素块与编码后JPG亮度图像

接下来是一组 640×480 分辨率的图像对比,图8(a)是原始RGB图像,大小为904 KB,图8(b)是过压缩后的JPEG图像,大小只有16.8 KB。



图8 RGB原始图像与JPG编码后图像

由测试结果验证,所作设计将原始RGB图像压缩编码为节省空间的JPG格式图像,达到了压缩编码的目的。与采用8 bit RGB格式的图像相比可以压缩到原大小的1/30左右。

本文基于FPGA实现了由BMP图像到JPEG图像的压缩编码过程。为了提高压缩效率、简化实现过程,在设计中对标准压缩算法做了些易于FPGA实现的优化,最终得到了较好的效果。并且将压缩编码器做成独立的IP模块,使之可以当作单独模块使用,简化了调用过程。

参考文献

- [1] 龚声荣,刘存平.数字图像处理与分析[M].北京:清华大学出版社,2008.
- [2] 徐光辉,程东旭,黄如.基于FPGA的嵌入式开发与应用[M].北京:电子工业出版社,2006.
- [3] 陈新华.EDA技术与应用[M].北京:机械工业出版社,2008.
- [4] ITU-T T.6, terminal equipment and protocols for telenmatic services[S].
- [5] 薛永林,刘珂,李凤亭.并行处理JPEG算法的优化[J].电子学报,2002(2):160-162.

(收稿日期:2012-05-08)

作者简介:

余磊,男,1987年生,硕士研究生,主要研究方向:数字专用集成电路。

李磊,男,1984年生,硕士研究生,主要研究方向:数字专用集成电路。

崔建明,男,1969年生,研究生导师,主要研究方向:数字专用集成电路。